

画像認識による物体の軌跡取得を用いたユーザインタフェース User-interface using tracks of an object

曾原 寿允†
Toshimitsu Sohara

堀 幸雄‡
Yukio Hori

今井 慈郎††
Yoshiro Imai

1. はじめに

近年、パソコンなどの情報機器や家電製品に対して画像認識技術を用いた様々な非接触型のインタフェースの提案が行われている。しかし、従来の方法では、モニター画面で動作を確認しながら操作を行うものが多く、利用する場面が限られてしまっている。そこで、本論文では画像認識による物体の軌跡取得を用いたユーザインタフェースを提案する。この方法では、物体の軌跡を用いてユーザの動きを方向ベクトルとして取得し、その動きを直接システムに反映するので、ユーザはモニター画面で動作を確認しなくても、直接的かつ体感的に操作を行うことが可能となる。

また、Intel社が提供するOpenCV(Open Source Computer Vision Library)[1]を活用することで、容易に画像処理を行うことが可能となっている。本研究では画像処理部分をOpenCVによる画像処理機能を用いてユーザインタフェースに組み込む方法でシステム開発を行っている。

2. システム概要と処理手順

開発したシステムのプロトタイプにおける処理手順は以下の(1)から(5)により実現される。

- (1) USBカメラにより画像を取得
 - (2) 追跡する物体の選択
 - (3) Cam-Shift法により対象を追跡し、座標情報の取得
 - (4) 対象の移動した量が所定の閾値を超えた場合、特定方向への移動と認識
 - (5) 取得情報をもとにシステムにフィードバック
- 図1にシステムの概要図を示す。

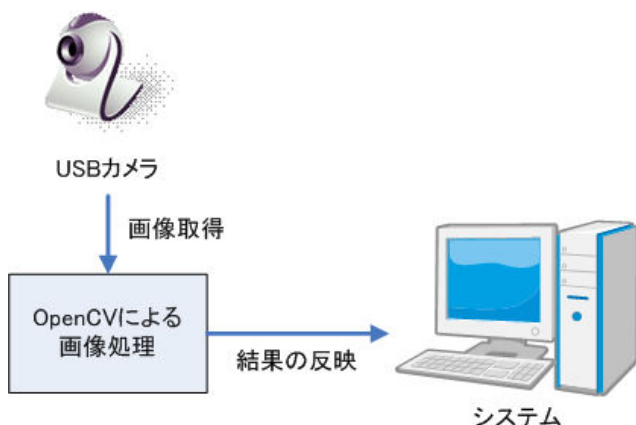


図1. システムの概要図

† 香川大学工学部

‡ 香川大学総合情報基盤センター

3. システムの流れ

3.1 追跡する物体の検知

追跡する物体を選択する方法を述べる。本システムでは、何も持たない手の動きを読み取りその動作を反映させることを前提としている。そこで、まず手の認識を行う。手の認識を行うにはカメラより得られた画像をHSV表色系に変換し、肌色領域の抽出を行う。HSVとは、ある色を色相、彩度、明度の3つの値で表す表色系である。カメラより得られた画像はRGBであるのでHSVに変換を行う。次に、HSV表色系によって抽出したHSV各成分に対して閾値を定め2値化を行う。2値化を行った際に欠損が発生することがあるため、画像の膨張と収縮を行い、欠損領域の補間を行う。これで肌色領域の抽出を行うことは可能となった。しかし、手以外の肌色領域も抽出されてしまう。そこで、画像中に手が大きく写っていることを前提とし、肌色領域の面積が最大の領域を手であるとして抽出を行う。

3.2 Convex Hullによる形状特徴抽出

本論文では対象を追跡するためにCamShift法を利用する。CamShift法では、まず追跡する対象を初期値として選択する必要がある。また、手がたまたまカメラの前を横切ったのか、それともユーザが意図的に操作を行おうとしているのかを区別する必要がある。そこで、Convex Hullを用いて手の形状抽出を行い、画像認識された手が「パー」の形から「グー」の形になったとき、それを選択し初期値とする。

Convex Hullとは、ある図形を含む最小の凸図形のことである。「図形内の任意の2つの画素を結ぶ線分が、この図形外の画素を通らないとき、この図形は凸である」というものである。Convex Hullを使用し、「グー」と「パー」に対する手の形状を抽出した例を図2に示す。図のように、手の領域を囲んだ線による図形がConvex Hullである。Convex Hullで囲んだ領域内に手がどれだけの面積を占めているかで、識別を行うことができる。例えば、パーの場合だと指と指の透き間が大きいので、Convex Hullで囲んだ領域中に占める手の領域の面積の割合は、グーよりも小さくなる。これを利用して、パーとグーの識別を行う。

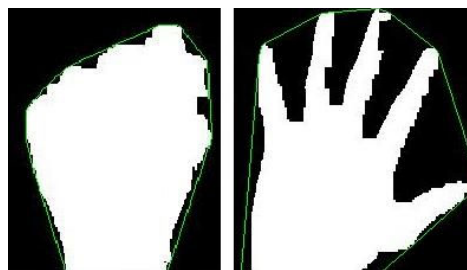


図2. Convex Hullによるグーとパーの認識

3.3 CamShift 法による追跡

CamShift 法は MeanShift 法を拡張したアルゴリズムである。MeanShift 法は、追跡対象領域における RGB の各ヒストグラムに着目し、現フレームの画像中で追跡対象のヒストグラム特徴により近くなる位置にシフトを行う。そして、シフトの移動量が一定以下になるか、繰り返す回数が上限に達するまで処理を繰り返すことにより、対象領域の現在の位置を追跡する方法である。MeanShift 法では、RGB のヒストグラム特徴を用いるのに対して、CamShift 法では色相値のヒストグラムを利用している。その結果、色情報が動的に変化する動画像に対してより安定した追跡を実現することが可能となる。

CamShift 法では始めに初期追跡領域を決定する必要があるため、3.2 節で述べたように Convex Hull を用いて手の形状抽出を行い、画像で認識された手がパーの形からグーの形になったときの肌色領域を初期値として選択する。

図 3 に CamShift 法の例を示す。

CamShift 法を実現するために、OpenCV で用意されている cvCamShift() 関数を用いる。この関数を用いて処理を行うと、対象の追跡結果の外接矩形が格納される構造体に対象の中心座標が格納される。この座標を用いて、対象物体がどの位置に移動したかを求めることができる。



図 3. CamShift 法による物体追跡の例

3.4 動作の判定方法

以上の方法により、物体の認識、追跡、追跡対象の座標の取得を行うことができた。次に、追跡対象の座標の変化をどのようにシステムに反映するかを述べる。

本システムでは、USB カメラより画像を取り込み、画像の解析を行うことにより、手の動きを取得しているので、「ブレ」が生じる危険性がある。そこで、物体追跡による物体の移動方向は上下左右の 4 種類とし、追跡対象の移動量が一定の閾値を超えた場合、その方向に移動したものと認識する。追跡対象の移動方向情報が得られたら、その情報を一度蓄積しておく。そして、その位置から再び物体の追跡を行う。この一連の流れをユーザが動作終了を示すまで繰り返す。

CamShift 法を行う時の初期値として、手の形状がパーからグーに変化したときに、初期値として選択を行ったが、手の形状がグーからパーに変化したときに動作終了であるとする。

そして、ストックされた物体の移動情報をシステムに反映する。最終的にシステムは得られた情報をもとになんらかの動作を行う。この動作はユーザによりカスタマイズ可能であるものとする。

図 4 に動作の判定方法のフローチャートを示す

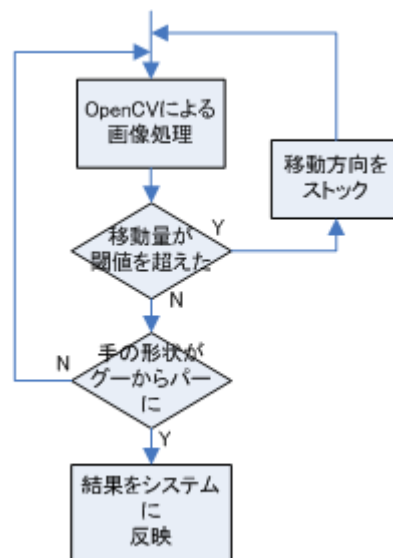


図 4. 動作の判定方法のフローチャート

4. システムの使用例と応用

本論文で提案したユーザインタフェースを用いて、ミュージックプレイヤーを操作するシステムを作成した。ここでは簡単のために、ミュージックプレイヤーは上下左右それぞれ一回の移動方向により操作可能なものとした。移動無しで「再生/停止」、上下方向の移動で「音量の増減」、右方向の移動で「次へ」そして、左方向の移動で「戻る」の動作を行うことができる。

次に、応用例について述べる。本論文で提案した方法を用いると、操作パネルを注視すると危険な場面、例えば車を運転中にカーステレオの操作を行う際など、において操作パネルを注視することなく操作を行うことが可能である。また、目の不自由な方がシステムに対して、何らかの操作を行いたい場面において、カメラの位置が既知であれば、スイッチやボタンなどを手探りで探さなくても、操作を行うことができると考えられる。

5. おわりに

本論文では、画像認識による物体の軌跡取得を用いたユーザインタフェースの提案を行った。また、OpenCV を用いたインタフェースの実現方法について説明した。

今後の課題として、認識精度の向上や、様々な環境への対応を考える必要がある。また、システム改良のため、定量的評価の一環として、複数ユーザによる動作チェックおよびアンケートを実施し、詳細なパラメータ調律を行い、改善すべき項目を明確化することなどを検討している。

【参考文献】

- [1] Open Source Computer Vision Library (OpenCV)
<http://www.intel.com/technology/computing/opencv/>
- [2] 奈良先端科学技術大学院大学 OpenCV プログラミングブック制作チーム著「OpenCV プログラミングブック」(2007年09月 毎日コミュニケーションズ)